DOI: 10.24996/ijs.2024.65.3.30

ISSN: 0067-2904

Numerical Solution of Linear Volterra Integral Equation of the Second Kind with Delay Using Lagrange Polynomials

Iman A. Dhari*, Muna M. Mustafa
Department of Mathematics, College of Science for Women, University of Baghdad, Baghdad, Iraq

Received: 30/11/2022 Accepted: 14/4/2023 Published: 30/3/2024

Abstract
In this study, the linear Volterra integral problem of the second kind will be treated with delay using a Lagrange polynomial. The Volterra integral problem is solved numerically using the chosen technique to obtain the best approximation. Additionally, the test examples are provided to demonstrate, through comparison with other methods' outcomes, the great degree of accuracy of the approximative solutions. Moreover, To verify the accuracy of the calculations that is used in these test examples, the absolute error is used to compare it to the exact solution. For this method, the program is written by MATLAB R2018a language.

Keywords: Delay Integral Equation, Lagrange Polynomial, Linear, Second Kind, Volterra Integral Equation.

الحل العددي لمعادلة فولتيرا التكاملية الخطية التباطؤية من النوع الثاني بأستخدام متعددات حدود لاكرانج

ايمان عبد الهادي ضاري* , منى منصور مصطفى
قسم الرياضيات، كلية العلوم للبنات، جامعة بغداد، بغداد، العراق

الخلاصة
في هذه الدراسة تم معالجة مسألة معادلة فولتيرا التكاملية الخطية التباطؤية من النوع الثاني باستخدام متعددة حدود لاكرانج. مسألة فولتيرا التكاملية تم حلها عدديا باستخدام التقنية المختارة لايجاد حل تقريبي افضل. بالاضافة الى ذلك تم اعطاء الامثلة الاختبارية لتوضيح الفرق بين الطريقة المقترحة ونتائج الطرق الاخرى من حيث درجة الدقة والتقريب للحل. تم الاعتماد على الخطأ المطلق للمقارنة به مع الحل المضبوط للتاكد من دقة حسابات هذه الأمثلة. كما وتم كتابة البرامج الخاصة بهذه الطريقة بأستخدام لغة الماتلاب.

1. Introduction

Scientists and researchers are interested in the delay integral equations because they can be found in a variety of mathematical formulations of modeling problems, for instance, in medical science, biomathematics, and models of population growth [1] , [2]. An essential subset of delay differential equations is the delay integral equations. Many methods for solving the delay integral equations have been discussed and investigated by various researchers. The variational iteration method was proposed to find the solution to linear and nonlinear delay Volterra

*Email: eman.abd@csw.uobaghdad.edu.iq

integral equations by [1] in 2012. The single-term Walsh series approach was used to solve delay Volterra integral equations [3]. The Z-decommission method has been applied to solve the equations of the Fredholm and Volterra types with time delay [4]. The non-polynomial spline is presented in [5] by A. M. Muhammad that is applied to the Volterra integral equation with delay. The computational Block-Pulse functions method was presented to find a numerical solution to Volterra integral equations with delay [6]. The Bernstein polynomials were proposed to present an effective solution for the 2nd kind of linear Volterra integral equations with delay [7]. The Nystrom-Clenshaw-Curtis quadrature was proposed in 2019 by [8] for solving Volterra integral equations with proportional delays. Also, in 2019 the finite difference method was suggested to solve an initial value problem for a linear first-order Volterra delay integro-differential equation [9] . The pseudo-spectral methods were suggested by [10] in 2020 to solve nonlinear Volterra integral equations with vanishing variable delays. An improved numerical scheme is proposed based on block pulse functions proposed to solve Volterra-type integral equations with time delay [11]. The sinc collocation method was used to solve the Volterra integral equations with proportional delay [12]. The Haar wavelet was developed for the solution of linear delay integral equations [13]. A class of the Volterra delays integral equations with noncompact operators is approximated by collocation methods [14] . The Haar collocation scheme is used for the solution to the class of system of delay integral equations for heterogeneous data communication[15] . Series solutions to the non-linear delay integral equations are considered by a modified approach of the homotopy analysis method [16]. Functional integral equations of the Volterra integral equations with constant delays type for significant test equations are investigated in [17] in 2022. Finally, a new approach to solving the linear fractional differential equation with a delay that uses the backward finite difference formula is discussed in [18] in 2022.

On the other hand, a lot of researchers have been utilizing Lagrange polynomials to obtain numerical solutions to various types of problems. In 2014, The authors of [19] and [20] used Lagrange polynomials to find a solution to integral and integro-differential Volterra-Fredholm Integral equations respectively, Also, in 2020, the author [21] used the Lagrange polynomials to solve linear fractional Volterra-Fredholm integro-differential equations.

In this work, we consider a linear Volterra integral equation of the 2nd kind with a constant time delay $\tau > 0$ of the form:

$$u(x) = \begin{cases} f(x) + \int_a^x k(\mathrm{x},\mathrm{t})u(t-\tau)\, dt\,, x \in [0,b], \tau \in (0,x) \\ \quad \varphi(x) \qquad\qquad\qquad\qquad\qquad\quad , x \in \ [-\tau, 0) \end{cases} \tag{1}$$

where the functions $f(x)$, $k(\mathrm{x},\mathrm{t})$, $\varphi(x)$are sufficiently smooth functions and $u(x)$ is the unknown function to be determined using the Lagrange polynomial.

This Volterra integral equation is population modeling for humans. The $u(x)$ is the number of population in time x and all children born at the time interval $0 < \tau < x$ who survive to time x. Also, $f(x)$ is the survival function which is the function of the number of people that survive to the age x [2].

This article is structured as follows: in Section 2, the definition of the Lagrange polynomial is given. Section 3 describes the methodology of the suggested method. Section 4 contains the algorithm of the method. Whilst section 5 contains the numerical test example. Finally, conclusions are highlighted in Section 6.

2. Lagrange Polynomial

The purpose of this part is to integrate the Lagrange polynomial's notations and definitions that have been given entirely in [21]:

For a set of *n*+1 data points $\{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$, define the Lagrange formula as follows:

$$P_n(x) = \sum_{j=0}^{n} u_j \; L_j(x) \;\; , \tag{2}$$

where

$$L_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{(x - x_k)}{(x_j - x_k)} \; .$$

3. Methodology

Below, a new technique for solving Eq.1 is provided by using a Lagrange polynomial. To apply the new technique, the interval $[-\tau, b]$ is first divided into 2N subintervals with equal space such that:

$$x_i = a + ih, i = 0, \pm 1, \pm 2, \dots, \pm N, \text{ where } x_{-N} = \tau, x_0 = a \; and \; x_N = b. \tag{3}$$

Now, applying the Lagrange polynomial in Eq.2 for the set of nodes that are defined in Eq.3, we get

$$P_{2N-1}(x) = \sum_{j=-N}^{N} u_j \; \prod_{\substack{k=-N \\ k \neq j}}^{N} \frac{(x - x_k)}{(x_j - x_k)} \;\; . \tag{4}$$

Which is a polynomial of degree 2N-1. Therefore, substituting Eq.4 in Eq.1 to obtain:

$$\sum_{j=-N}^{N} u(x_j) \; \prod_{\substack{k=-N \\ k \neq j}}^{N} \frac{(x - x_k)}{(x_j - x_k)} \; = f(x) + \int_a^x k(\mathrm{x}, \mathrm{t}) \, (\sum_{j=-N}^{N} u(x_j) \; \prod_{\substack{k=-N \\ k \neq j}}^{N} \frac{(x - \tau - x_k)}{(x_j - x_k)}) \, . \tag{5}$$

Note that $u(x_j) = \varphi(x_j) \, , j = -N, -N+1, \dots, -1.$

To find the solution to *u(x)* at the point $x_j, j = 0,1, \dots, N$, substituting x=x_j in Eq.5 to get a system of $N + 1$ equations, which is given by:

$$A\vec{u} = \vec{b} \tag{6}$$

Where $A = [a_{ij}]$, $\vec{u} = [u_i]$, and $\vec{b} = [b_i]$, such that:

$$a_{ij} = \begin{cases} 1 - \int_a^{x_i} k(x_i, t) \, L_{i,j}(t - \tau) \, dt \;\; if \; i = j \\ -\int_a^{x_i} k(x_i, t) L_{i,j}(t - \tau) \, dt \;\; if \; i \neq j \end{cases} \;\; , \quad i, j = 0,1, \dots, N \;\; . \tag{7}$$

$$b_i = f(x_i) + \sum_{j=-N}^{-1} u(x_j) \int_a^{x_i} k(x_i, t) \, L_{i,j}(t) \, dt, \quad i = 0,1, \dots, N \, . \tag{8}$$

such that

$$L_{i,j}(x) = \prod_{\substack{j=-N \\ j \neq i}}^{N} \frac{(x - x_j)}{(x_i - x_j)} \;\; . \tag{9}$$

i. e.

$$\begin{bmatrix} 1 - \int\limits_a^{x_0} k(x_0, t) \, L_{0,0}(t - \tau) \, dt & -\int\limits_a^{x_0} k(x_0, t) \, L_{0,1}(t - \tau) \, dt & -\int\limits_a^{x_0} k(x_0, t) \, L_{0,2}(t - \tau) \, dt \cdots & \cdots & -\int\limits_a^{x_0} k(x_0, t) \, L_{0,N}(t - \tau) \, dt \\ -\int\limits_a^{x_1} k(x_1, t) \, L_{1,0}(t - \tau) \, dt & 1 - \int\limits_a^{x_1} k(x_1, t) \, L_{1,1}(t - \tau) \, dt & -\int\limits_a^{x_1} k(x_1, t) \, L_{1,2}(t - \tau) \, dt \cdots & \cdots & -\int\limits_a^{x_1} k(x_1, t) \, L_{1,N}(t - \tau) \, dt \\ -\int\limits_a^{x_2} k(x_2, t) \, L_{2,0}(t - \tau) \, dt & -\int\limits_a^{x_2} k(x_2, t) \, L_{2,1}(t - \tau) \, dt & 1 - \int\limits_a^{x_2} k(x_2, t) \, L_{2,2}(t - \tau) \, dt \cdots & \cdots & -\int\limits_a^{x_2} k(x_2, t) \, L_{2,N}(t - \tau) \, dt \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\int\limits_a^{x_N} k(x_N, t) \, L_{N,0}(t - \tau) \, dt & -\int\limits_a^{x_N} k(x_N, t) \, L_{N,1}(t - \tau) \, dt & -\int\limits_a^{x_N} k(x_N, t) \, L_{N,2}(t - \tau) \, dt \cdots & \cdots & 1 - \int\limits_a^{x_N} k(x_N, t) \, L_{N,N}(t - \tau) \, dt \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_N \end{bmatrix}$$

$$= \begin{bmatrix} f(x_0) + \sum_{j=-N}^{-1} u(x_j) \int_a^{x_0} k(x_0, t) \, L_{0,j}(t) dt \\ f(x_1) + \sum_{j=-N}^{-1} u(x_j) \int_a^{x_1} k(x_1, t) \, L_{1,j}(t) dt \\ f(x_2) + \sum_{j=-N}^{-1} u(x_j) \int_a^{x_2} k(x_2, t) \, L_{2,j}(t) dt \\ \vdots \\ \vdots \\ f(x_N) + \sum_{j=-N}^{-1} u(x_j) \int_a^{x_N} k(x_N, t) \, L_{N,j}(t) dt \end{bmatrix}$$

4. Description of the Algorithm

In this algorithm, the Lagrange polynomial is used to find the numerical solution to the linear Volterra integral equation of the 2nd kind with delay.:

Input: $f(x), k(x,t)$ *and* $\varphi(x)$ (the functions are defined in Eq.1), a, b *and* τ (the constants are defined in Eq.1), n is the number of subintervals in $[a,b]$, such that $N = 2n$.

Output: $\vec{u} = [u_i]$ (the vector is defined in Eq.6)
Step 1: Set $h = \frac{b-a}{n}$, $N \in \mathbb{N}$.
Step 2: Calculate $x_i = a + ih$, with $x_0 = a$ and $x_n = b$, $i = 0,1,2, \dots, n$.
Step 3: Use steps 1 and 2 with Eq.7 to find the matrix a.
Step 4: Compute the vector b using Eq.8 and steps 1 and 2.
Step 5: Find the solution to the linear system in Eq.6 using the above steps and the Gauss elimination method.
Moreover, an integral part of Eq. 7 and Eq. 8 was calculated using the exact solution for integral instead of numerical integration.

5. Numerical Test Example

In this section, to illustrate the proposed technique for solving the linear Volterra integral equation of the 2nd kind with delay, some numerical test examples are given. The exact solution is known and used to demonstrate the validity of the numerical solution that is produced by using our method. Additionally, at all points in these examples, the absolute error is determined which is defined by the following:

$$Absolute\ error = |exact\ solution - numerical\ solution|$$

Test Example 1

Consider the following linear Volterra integral equation of the second kind with delay[11]:

$$u(x) = \begin{cases} \sin x + x^2 \cos(x-1) - x^2 \cos(-1) + \int_0^x x^2 u(t-1)\, dt, & x \in [0,1] \\ x - \frac{x^3}{3!}, & x \in [-1,0) \end{cases}$$

with the exact solution being $u(x) = \sin(x)$, for $x \in [0,1]$. Table 1 contains the results of this example using the Lagrange polynomial to get a numerical solution in the range [0, 1], with *n=10* and *h=0.1*.

Table 1: The numerical solution and the exact solution with the absolute error of test example 1

n	Numerical solution	Exact solution	Absolute error
0	-4.25077117680179e-18	0	4.25077117680179e-18
0.1	0.0998386935148979	0.099833416646828	5.27686806976979e-06
0.2	0.198705477158925	0.198669330795061	3.61463638635105e-05
0.3	0.295619245147809	0.295520206661340	9.90384864697202e-05
0.4	0.389610068370245	0.389418342308651	0.00019172606159443
0.5	0.479735785214625	0.479425538604203	0.000310246610422176
0.6	0.565094976138383	0.564642473395035	0.000452502743348085
0.7	0.644835869294776	0.644217687237691	0.00061818205708449
0.8	0.718164110192649	0.717356090899523	0.000808019293126157
0.9	0.784349621738702	0.783326909627483	0.00102271211121854
1	0.842733601377288	0.841470984807897	0.00126261656939119

Compared to the results in [11] which solve the same examples using block pulse functions, it is clear that the results obtained using the Lagrange polynomial represent the best approximation.

Test Example 2

Consider another linear Volterra integral equation of the second kind with delay [11][16]:

$$u(x) = \begin{cases} e^x + xe^{x-1} + xe^{-1} + \int_0^x xu(t-1)\,dt \ , & x \in [0,1] \\ e^x, & x \in [-1,0) \end{cases}$$

with the exact solution being $u(x) = \mathrm{e}^{\mathrm{x}}$, for $x \in [0,1]$. Table 2 contains the results of this example's use of the Lagrange polynomial to get a numerical solution in the range [0, 1], with *n=10* and *h=0.1*.

Table 2: The numerical solution and the exact solution with the absolute error of test example 2

$\boldsymbol{n}$	Numerical solution	Exact solution	Absolute error
0	1	1	0
0.1	1.10517091807565	1.10517091807565	0
0.2	1.22140275816017	1.22140275816017	0
0.3	1.349858807576	1.349858807576	0
0.4	1.49182469764127	1.49182469764127	0
0.5	1.64872127070013	1.64872127070013	0
0.6	1.82211880039051	1.82211880039051	0
0.7	2.01375270747048	2.01375270747048	0
0.8	2.22554092849247	2.22554092849247	0
0.9	2.45960311115695	2.45960311115695	0
1	2.71828182845905	2.71828182845905	0

Compared to the results in [11] which solve the same examples using block pulse functions and the results obtained in [16] using series solutions, it is clear that the results obtained using the Lagrange polynomial represent the best approximation.

Test Example 3

Consider another linear Volterra integral equation of the second kind with a delay[6]

$$u(x) = \begin{cases} x^2\left(1-\frac{1}{2}\right) + \frac{2x^3}{3} - \frac{x^4}{4} + \int_0^x xu(t-1)\,dt & ,x \in [0,1] \\ x^2 & ,\ x \in [-1,0) \end{cases}$$

with the exact solution being $u(x) = \mathrm{x}^2$, for $x \in [0,1]$. Table 3 contains the results of this example using the Lagrange polynomial to get a numerical solution in the range [0, 1], with *n=10* and *h=0.1*.

Table 3: The numerical solution and the exact solution with the absolute error of test example 3

$\boldsymbol{n}$	Numerical solution	Exact solution	Absolute error
0	1.97872909775964e-17	0	1.97872909775964e-17
0.1	0.0100000000000001	0.01	0
0.2	0.0400000000000001	0.04	0
0.3	0.0900000000000001	0.09	0
0.4	0.16	0.16	0
0.5	0.25	0.25	0
0.6	0.36	0.36	0
0.7	0.49	0.49	0
0.8	0.64	0.64	0
0.9	0.81	0.81	0
1	1	1	0

Compared to the results in [6] which solve the same examples using the block-pulse function approach, it is clear that the results obtained using the Lagrange polynomial represent the best approximation as seen in Table 4.

Table 3: comparison with the results in [6]

x	Exact solution	Numerical solution using Lagrange Polynomial	Results obtained in [6]
0	0	1.97872909775964e-17	0.001
0.1	0.01	0.0100000000000001	0.0108
0.2	0.04	0.0400000000000001	0.391
0.3	0.09	0.0900000000000001	0.0914
0.4	0.16	0.16	0.1577
0.5	0.25	0.25	0.2558
0.6	0.36	0.36	0.3634
0.7	0.49	0.49	0.4807
0.8	0.64	0.64	0.6493
0.9	0.81	0.81	0.8061

6. Conclusions:

In this work, the Lagrange polynomial has been applied to evaluate the second kind of linear Volterra integral equation with delay. The following points are suggested based on the numerical findings that are got from the previous examples:

- The approximations developed by MATLAB software show the accuracy and validity of the proposed approach.
- The method can be improved and applied to the nonlinear Volterra integral equations.
- The approach can be extended to solve the nth-order nonlinear Volterra integro-differential problem.
- Use another type of the Lagrange polynomial for example modified Lagrange polynomial.

7. conflict of interest

Conflict of Interest: The authors declare that they have no conflicts of interest.

References

[1] M. Avaji, J. S. Hafshejani, S. S. Dehcheshmeh, and D. F. Ghahfarokhi, “Solution of Delay Volterra Integral Equations using the Variational Iteration Method,” *Journal of Applied Sciences*, vol. 12, no. 2. pp. 196–200, 2012.

[2] M. Nouri and K. Maleknejad, “Numerical Solution of Delay Integral Equations by Using Block Pulse Functions Arises in Biological Sciences,” *International Journal of Mathematical Modelling & Computations*, vol. 06, no. 03, pp. 221–232, 2016.

[3] R. C. G. Sekar and K. Murugesan, “Numerical Solutions of Delay Volterra Integral Equations Using Single-term Walsh Series Approach,” *International Journal of Applied and Computational Mathematics*, vol. 3, no. 3, pp. 2409–2421, 2017.

[4] M. Almousa and O. N. Ogilat, “A new approach for solution of linear time-delay integral equations via Z-decommission method,” *Far East Journal of Mathematical Sciences*, vol. 102, no. 11, pp. 2551–2561, 2017.

[5] A. M. Muhammad, “Numerical Solution Of Volterra Integral Equation With Delay By Using Non-Polynomial Spline Function,” *Misan Journal for Academic Studies*, p. 133, 2017.

[6] H. K. Dawood, “Computational Block-Pulse Functions Method for Solving Volterra Integral Equations with Delay,” *JOURNAL OF UNIVERSITY OF BABYLON for Pure and Applied Sciences*, vol. 27, no. 1, pp. 32–42, 2019.

[7] A. M. Muhammad and A. M. Ayal, “Numerical Solution of Linear Volterra Integral Equation with Delay using Bernstein Polynomial,” *International Electronic Journal of Mathematics Education*, vol. 14, no. 3, pp. 735–740, 2019.
[8] W. L. Guo and F. R. Lin, “Nyström-Clenshaw-Curtis quadrature for the solution of Volterra integral equations with proportional delays,” *AIP Conference Proceedings*, vol. 2116, no. July, 2019.
[9] E. Cimen and S. Yatar, “Numerical solution of volterra integro-differential equation with delay,” *Journal of Mathematics and Computer Science*, vol. 20, no. 3, pp. 255–263, 2019.
[10] Z. Xiao-yong, “A new strategy for the numerical solution of nonlinear Volterra integral equations with vanishing delays,” *Applied Mathematics and Computation*, vol. 365, p. 124608, 2020.
[11] S. S. C., H. B. S., and L. Lata, “An improved method based on block pulse functions for the numerical solution of Volterra type delay integral equations,” *Malaya Journal of Matematik*, vol. S, no. 1, pp. 163–167, 2020.
[12] B. Mallick and P. K. Sahu, “A Numerical Approach for Solving Volterra Integral Equation with Proportional Delay using Sinc-Collocation Method,” *International Journal of Applied and Computational Mathematics*, vol. 6, no. 5, 2020.
[13] R. Amin, K. Shah, M. Asif, and I. Khan, “Efficient numerical technique for solution of delay Volterra-Fredholm integral equations using Haar wavelet,” *Heliyon*, vol. 6, no. 10, p. e05108, 2020.
[14] H. Song, Y. Xiao, and M. Chen, “Collocation methods for third-kind Volterra integral equations with proportional delays,” *Applied Mathematics and Computation*, vol. 388, no. 11771111, p. 125509, 2021.
[15] H. Wu, R. Amin, A. Khan, S. Nazir, and S. Ahmad, “Solution of the Systems of Delay Integral Equations in Heterogeneous Data Communication through Haar Wavelet Collocation Approach,” *Complexity*, vol. 2021, 2021.
[16] S. N. Huseen and A. S. Tayih, “Series Solutions of Delay Integral Equations via a Modified Approach of Homotopy Analysis Method,” *Iraqi Journal of Science*, vol. 62, no. 11, pp. 4006–4018, 2021.
[17] P. Darania and S. Pishbin, “Stability Analysis of the High-order Multistep Collocation Method for the Functional Integral Equations with Constant Delays,” *Boletim da Sociedade Paranaense de Matematica*, vol. 40, no. 40, pp. 1–11, 2022.
[18] A. K. Hameed and M. M. Mustafa, “Numerical Solution of Linear Fractional Differential Equation with Delay Through Finite Difference Method,” *Iraqi Journal of Science*, vol. 63, no. 3, pp. 1232–1239, 2022.
[19] M. M. Mustafa and I. N. Ghanim, “Numerical Solution of Linear Volterra-Fredholm Integral Equations Using Lagrange Polynomials,” vol. 4, no. 5, 2014.
[20] M. M. Mustafa and A. M. Muhammad, “Numerical Solution of Linear Volterra-Fredholm Integro-Differential Equations Using Lagrange Polynomials,” vol. 4, no. 9, 2014.
[21] N. K. Salman and M. M. Mustafa, “Numerical solution of fractional Volterra-Fredholm integro-differential equation using lagrange polynomials,” *Baghdad Science Journal*, vol. 17, no. 4, pp. 1234–1240, Dec. 2020.